CLAIMS:

1. A system for adaptive detection of streamed packet loss by a receiver of a plurality of streamed packets transmitted over a network from a given sender to the receiver, comprising:

a circular buffer (10) of size $m>1$ entries (13) , each entry (13) having at most one sequence number (15) of a streamed packet that has not been received and is possibly lost;

a packet loss detection module (33) that uses the circular buffer (10) to detect and store therein a sequence number (15) of a non-received and possibly lost packet, to detect therein and remove therefrom a sequence number (15) of a lost packet and declare the packet lost, and to remove therefrom a sequence number (15) of a possibly lost packet that is received from the given sender;

an adaptation module (37) that adapts the system to a network condition,

wherein a loss detection latency is determined by the size m of the circular buffer (10) and the loss declaration is possibly false.

2. The system of claim 1, wherein m is initially set to 4.

3. The system of claim 1, further comprising:

a variable s having an initial value of 1 and being adapted to store a highest sequence number of a streamed packet transmitted over the network from the given sender and received by the receiver;

a pointer *P (12)* having an initial position pointing at a pre-determined location in the circular chain and being adapted to circulate sequentially through the *m* entries of said circular buffer (10) beginning at an entry (13) in the circular buffer (10) that is next in sequence to the entry (13) corresponding to the variable *s*;

wherein,

for a streamed packet received from the given sender, the packet loss detection module (33) checks the sequence number of the received packet against the variable *s* and performs one of the following -

a. if a hole in the sequence of received packets is observed beginning at the location pointed at by the pointer *P (12)*, each entry (13) of the circular buffer (10) that is in the hole is checked for a sequence number (15) of a possibly lost packet and the corresponding packet is declared lost, a total of declared losses *declared_losses* is increased by one, each sequence number in the hole is stored in ascending order in a sequential entry (13) beginning at the location pointed at by the pointer *P (12)*, *P (12)* is updated to point to the entry (13) in the circular buffer (10) following hole, and *s* is set equal to the sequence number of the received pkt,

b. if a hole in the sequence of received packets in not observed the entry (13) pointed at by the pointer *P (12)* is checked for a sequence number (15) of a possibly lost packet and the corresponding packet is declared lost, a total of declared losses *declared_losses* is increased by one, the entry (13) is cleared, *P (12)* is updated to point to the next entry (14) in the circular buffer (10), and *s* is set equal to the sequence number of the received pkt,

c. if an out of order packet is observed, the entries of the circular buffer (10) are searched to find one that contains a sequence number (15) equal to the sequence

number of the received packet and if found the entry (13) is cleared, if not found a false declaration rate *false_declared_losses* is increased by one.

4. The system of claim 3, wherein:

the network condition is at least one of a success rate of transmission (*success_rate)* and the false declaration rate (*false_rate)* wherein the *success_rate* is initially set to a pre-determined expected rate (*EXPECT_RATE)*; and

the adaptation module (37) adjusts the size *m* of the circular buffer (10) according to the network condition as follows

a. *m* is increased if *false_rate* > *TOLERABLE_RATE* where *TOLERABLE_RATE* is a predetermined threshold and an entry (13) is added to the circular buffer (10), or

b. *m* is decreased if

$$success\_rate = \frac{declared\_losses - falsely\_declared\_losses}{EXPECT\_RATE} > EXPECT\_RATE$$

and an entry (13) is removed from the circular buffer (10).

5. The system of claim 4, wherein the circular buffer (10) is a circular buffer (10) chain

Bi for *i=1, ..., m* of a plurality of *m>1* buffers such that each of said plurality of buffers is an entry (13) comprising a pointer to the next buffer (14) in the chain and a value for storing a sequence number (15) of a non-received buffer and the pointer *P (12)* point to a buffer in the chain.

6. A system for adaptive detection of streamed packet loss by a receiver of a plurality of streamed packets transmitted over a network from a given sender to the receiver, comprising:

a circular buffer (10) of size $m>1$ entries, each entry (13) having at most one sequence number (15) of a streamed packet that has not been received and is possibly lost;

a packet loss detection module (33) that uses the circular buffer (10) to detect and store therein a sequence number (15) of a non-received and possibly lost packet, to detect therein and remove therefrom a sequence number (15) of a lost packet and declare the packet lost, and to remove therefrom a sequence number (15) of a possibly lost packet that is received from the given sender;

means for adapting the system to a network condition (37),

wherein a loss detection latency is determined by the size m of the circular buffer (10) and the loss declaration is possibly false.

7. The system of claim 6, further comprising:

a variable s having an initial value of 1 and being adapted to store a highest sequence number of a streamed packet transmitted over the network from the given sender and received by the receiver;

a pointer *P (12)* having an initial position pointing at a pre-determined location in the circular chain and being adapted to circulate sequentially through the m entries of said circular buffer (10) beginning at an entry (13) in the circular buffer (10) that is next in sequence to the entry (13) corresponding to the variable s;

wherein,

for a streamed packet received from the given sender, the packet loss detection module (33) checks the sequence number of the received packet against the variable *s* and performs one of the following -

a. if a hole in the sequence of received packets is observed beginning at the location pointed at by the pointer *P (12)*, each entry (13) of the circular buffer (10) that is in the hole is checked for a sequence number (15) of a possibly lost packet and the corresponding packet is declared lost, a total of declared losses *declared_losses* is increased by one, each sequence number in the hole is stored in ascending order in a sequential entry (13) beginning at the location pointed at by the pointer *P (12)*, *P (12)* is updated to point to the entry (13) in the circular buffer (10) following hole, and *s* is set equal to the sequence number of the received pkt,

b. if a hole in the sequence of received packets in not observed the entry (13) pointed at by the pointer *P (12)* is checked for a sequence number (15) of a possibly lost packet and the corresponding packet is declared lost, a total of declared losses *declared_losses* is increased by one, the entry (13) is cleared, *P (12)* is updated to point to the next entry (14) in the circular buffer (10), and *s* is set equal to the sequence number of the received pkt,

c. if an out of order packet is observed, the entries (13) of the circular buffer (10) are searched to find one that contains a sequence number (15) equal to the sequence number of the received packet and if found the entry (13) is cleared, if not found a false declaration rate *false_declared_losses* is increased by one.

8. A method for adaptive detection of streamed packet loss by a receiver of a plurality of streamed packets transmitted over a network from a given sender to the receiver, comprising the steps of:

providing a circular buffer (10) of size $m>1$ entries, each entry (13) having at most one sequence number (15) of a streamed packet that has not been received and is possibly lost;

receiving from the given sender a streamed packet having a sequence number;

using the circular buffer (10) and the sequence number of the received packed to perform one of the steps of:

a. detecting and storing in the circular buffer (10) a sequence number (15) of a non-received and possibly lost packet,

b. detecting in the circular buffer (10) and removing therefrom a sequence number (15) of a lost packet and declaring the packet lost such that the loss declaration is possibly false, and

c. removing from the circular buffer (10) a sequence number (15) of a possibly lost packet that corresponds to the sequence number of the received packet;

adapting the method to a network condition such that a loss detection latency is determined by the size m of the circular buffer (10).

9. The method of claim 8, further comprising the steps of:

providing a variable s having an initial value of 1;

setting the provided variable s = max(s, sequence number of the received streamed packet);

providing a pointer *P (12)* having an initial position pointing at a pre-determined location in the circular buffer (10) that is adapted to circulate sequentially through the *m* entries of the provided circular buffer (10) beginning at an entry (13) in the circular buffer (10) that is next in sequence to the entry (13) corresponding to the variable *s*;

for a streamed packet received from the given sender, checking the sequence number of the received packet against the variable *s* and performing one of the following steps-

a. if a hole in the sequence of received packets is observed beginning at the location pointed at by the pointer *P (12)*,

a.1 checking each entry (13) of the circular buffer (10) that is in the hole for a sequence number (15) of a possibly lost packet and declaring the corresponding packet lost,

a.2 if a packet is declared lost, increasing a total of declared losses *declared_losses* by one,

a.3 storing each sequence number in the hole in ascending order in a sequential entry (13) beginning at the location pointed at by the pointer *P (12)*,

a.4 updating the pointer *P (12)* to point to the entry (13) in the circular buffer (10) following hole, and

a.5 setting *s* equal to the sequence number of the received pkt;

b. if a hole in the sequence of received packets in not observed

b.1 checking the entry (13) pointed at by the pointer *P (12)* for a sequence number (15) of a possibly lost packet and declaring the corresponding packet lost,

b2. if a packet is declared lost, increasing a total declared losses *declared_losses* by one,

b.3 clearing the entry (13) pointed at by *P (12)*,

b.4 updating *P (12)* to point to the next entry (14) in the circular buffer (10), and

b.5 setting *s* equal to the sequence number of the received pkt;

c. if an out of order packet is observed

c.1 searching the entries (13) of the circular buffer (10) to find one that contains a sequence number (15) equal to the sequence number of the received packet,

c.2 if found, clearing the entry (13),

c.3 if not found, increasing a false declaration rate *falsely_declared_losses* by one.

10. The method of claim 9, wherein:

the network condition is at least one of a success rate of transmission (*success_rate)* and the false declaration rate (*false_rate))* wherein the *success_rate* is initially set to a pre-determined expected rate (*EXPECT_RATE)*; and

the adaptation step adjusts the size *m* of the circular buffer (10) according to the network condition by performing one of the following steps:

d. if *false_rate* > *TOLERABLE_RATE* where *TOLERABLE_RATE* is a predetermined

threshold performing the following steps

d.1 increasing *m* by 1, and

d.2 adding an entry (13) to the circular buffer (10), or

e. if *success_rate* > *EXPECT_RATE*

e.1 decreasing *m* by 1,

e.2 removing an entry (13) from the circular buffer (10),

11. The method of claim 9, wherein *m* is initially set to 4.

12. A computer program product for use in conjunction with a processor to adapt detection of streamed packet loss by a receiver of a plurality of streamed packets transmitted over a network from a given sender to the receiver, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

a circular buffer (10) of size *m>1* entries, each entry (13) having at most one sequence number (15) of a streamed packet that has not been received and is possibly lost;

a packet loss detection routine (33) including instructions for using the circular buffer (10) to detect and store therein a sequence number (15) of a non-received and possibly lost packet, detect therein and remove therefrom a sequence number (15) of a lost packet and declare the packet lost, and remove therefrom a sequence number (15) of a possibly lost packet that is received from the given sender;

an adaptation routine (37) including instructions that adapt the system to a network condition,

wherein a loss detection latency is determined by the size *m* of the circular buffer (10) and the loss declaration is possibly false.

13. The computer program product of claim12, further comprising:

a variable *s* having an initial value of 1 and being adapted to store a highest sequence number of a streamed packet transmitted over the network from the given sender and received by the receiver;

a pointer *P (12)* having an initial position pointing at a pre-determined location in the circular chain and being adapted to circulate sequentially through the *m* entries of said circular buffer (10) beginning at an entry (13) in the circular buffer (10) that is next in sequence to the entry (13) corresponding to the variable *s*;

wherein,

for a streamed packet received from the given sender, the instructions of the packet loss detection routine (33) check the sequence number of the received packet against the variable *s* and perform one of the following -

a. if a hole in the sequence of received packets is observed beginning at the location pointed at by the pointer *P (12)*, each entry (13) of the circular buffer (10) that is in the hole is checked for a sequence number (15) of a possibly lost packet and the corresponding packet is declared lost, total of declared losses *declared_losses* is increased by one, each sequence number in the hole is stored in ascending order in a sequential entry (13) beginning at the location pointed at by the pointer *P (12)*, *P (12)* is updated to point to the entry (13) in the circular buffer (10) following hole, and *s* is set equal to the sequence number of the received pkt,

b. if a hole in the sequence of received packets in not observed the entry (13) pointed at by the pointer *P (12)* is checked for a sequence number (15) of a possibly lost packet and the corresponding packet is declared lost, a total of declared

losses *declared_losses* is increased by one, the entry (13) is cleared, *P (12)* is updated to point to the next entry (14) in the circular buffer (10), and *s* is set equal to the sequence number of the received pkt,

c. if an out of order packet is observed, the entries (13) of the circular buffer (10) are searched to find one that contains a sequence number (15) equal to the sequence number of the received packet and if found the entry (13) is cleared, if not found a false declaration rate *falsely_declared_losses* is increased by one and the total of *declared losses declared_losses* is decreased by one.

14. The computer program product of claim 13, wherein:

the network condition is at least one of a success rate of transmission (*success_rate)* and a false declaration rate *(false)rate)* wherein the *success_rate* is initially set to a pre-determined expected rate (*EXPECT_RATE)*; and

the adaptation routine (37) adjusts the size *m* of the circular buffer (10) according to the network condition as follows

a. *m* is increased if *false_rate* > *TOLERABLE_RATE* where *TOLERABLE_RATE* is a predetermined threshold and an entry (13) is added to the circular buffer (10), or

b. *m* is decreased if
success_rate =
$$\frac{declared\_losses\text{-}falsely\_declared\_losses}{EXPECT\_RATE} > EXPECT\_RATE$$ and an entry (13) is removed from the circular buffer (10).